Your search matched **11** of **1051129** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or entering a new one in the text box.

analyzer<and>control flow Search

☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

---

1 **Control flow checking in pipelined RISC microprocessors: the Motorola MC88100 case study**
Delord, X.; Saucier, G.;
Real Time, 1990. Proceedings., Euromicro '90 Workshop on , 6-8 June 1990
Pages:162 - 169

[Abstract] [PDF Full-Text (592 KB)] **IEEE CNF**

---

2 **Bounding pipeline and instruction cache performance**
Healy, C.A.; Arnold, R.D.; Mueller, F.; Whalley, D.B.; Harmon, M.G.;
Computers, IEEE Transactions on , Volume: 48 , Issue: 1 , Jan. 1999
Pages:53 - 70

[Abstract] [PDF Full-Text (772 KB)] **IEEE JNL**

---

3 **Bounding worst-case instruction cache performance**
Arnold, R.; Mueller, F.; Whalley, D.; Harmon, M.;
Real-Time Systems Symposium, 1994., Proceedings. , 7-9 Dec. 1994
Pages:172 - 181

[Abstract] [PDF Full-Text (876 KB)] **IEEE CNF**

---

4 **A priori execution time analysis for parallel processes**
Halang, W.A.;
Real Time, 1989. Proceedings., Euromicro Workshop on , 14-16 June 1989
Pages:62 - 65

[Abstract] [PDF Full-Text (312 KB)] **IEEE CNF**

---

5 **Variable precision reaching definitions analysis for software**

maintenance
Tonella, P.; Antoniol, G.; Fiutem, R.; Merlo, E.;
Software Maintenance and Reengineering, 1997. EUROMICRO 97., First Euromicro Conference on , 17-19 March 1997
Pages:60 - 67

[Abstract] [PDF Full-Text (688 KB)] **IEEE CNF**

---

6 **A prototype system for static and dynamic program understanding**
Olshefski, D.P.; Cole, A.;
Reverse Engineering, 1993., Proceedings of Working Conference on , 21-23 May 1993
Pages:93 - 106

[Abstract] [PDF Full-Text (2664 KB)] **IEEE CNF**

---

7 **Visual knowledge engineering**
Eisenstadt, M.; Domingue, J.; Rajan, T.; Motta, E.;
Software Engineering, IEEE Transactions on , Volume: 16 , Issue: 10 , Oct. 1990
Pages:1164 - 1177

[Abstract] [PDF Full-Text (1420 KB)] **IEEE JNL**

---

8 **Impact analysis and change management for avionics software**
Loyall, J.P.; Mathisen, S.A.; Satterthwaite, C.P.;
Aerospace and Electronics Conference, 1997. NAECON 1997., Proceedings of the IEEE 1997 National , Volume: 2 , 14-17 July 1997
Pages:740 - 747 vol.2

[Abstract] [PDF Full-Text (1476 KB)] **IEEE CNF**

---

9 **Integrating the timing analysis of pipelining and instruction caching**
Healy, C.A.; Whalley, D.B.; Harmon, M.G.;
Real-Time Systems Symposium, 1995. Proceedings., 16th IEEE , 5-7 Dec. 1995
Pages:288 - 297

[Abstract] [PDF Full-Text (944 KB)] **IEEE CNF**

---

10 **A hybrid program knowledge base for static program analyzers**
Jarzabek, S.; Han Shen; Hock Chuan Chan;
Software Engineering Conference, 1994. Proceedings., 1994 First Asia-Pacific , 7-9 Dec. 1994
Pages:400 - 409

[Abstract] [PDF Full-Text (684 KB)] **IEEE CNF**

---

11 **Ada source code analysis for automatic test case generation**
Santhanam, V.;
AUTOTESTCON '92. IEEE Systems Readiness Technology Conference, Conference Record , 21-24 Sept. 1992
Pages:325

[Abstract] [PDF Full-Text (24 KB)] **IEEE CNF**

Terms used **analyzer vertex 'control flow'**

Found **27** of **139,567**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

☐ Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 21 - 27 of 27

Result page: previous 1 **2**

Relevance scale

21 Beyond induction variables

Michael Wolfe

July 1992 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1992 conference on Programming language design and implementation**, Volume 27 Issue 7

Full text available: pdf(1.20 MB) Additional Information: full citation, abstract, references, citings, index terms

Induction variable detection is usually closely tied to the strength reduction optimization. This paper studies induction variable analysis from a different perspective, that of finding induction variables for data dependence analysis. While classical induction variable analysis techniques have been used successfully up to now, we have found a simple algorithm based on the Static Single Assignment form of a program that finds all linear induction variables in a loop. Moreover, this algorith ...

22 Visualization for program understanding: A system for graph-based visualization of the evolution of software

Christian Collberg, Stephen Kobourov, Jasvir Nagra, Jacob Pitts, Kevin Wampler

June 2003 **Proceedings of the 2003 ACM symposium on Software visualization**

Full text available: pdf(3.80 MB) Additional Information: full citation, abstract, references

We describe GEVOL, a system that visualizes the evolution of software using a novel graph drawing technique for visualization of large graphs with a temporal component. GEVOL extracts information about a Java program stored within a CVS version control system and displays it using a temporal graph visualizer. This information can be used by programmers to understand the evolution of a legacy program: Why is the program structured the way it is? Which programmers were responsibl ...

23 A general-purpose algorithm for analyzing concurrent programs

Richard N. Taylor

May 1983 **Communications of the ACM**, Volume 26 Issue 5

Full text available: pdf(1.61 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Developing and verifying concurrent programs presents several problems. A static analysis algorithm is presented here that addresses the following problems: how processes are synchronized, what determines when programs are run in parallel, and how errors are detected in the synchronization structure. Though the research focuses on Ada, the results can be applied to other concurrent programming languages such as CSP.

24 Visualizing interactions in program executions

Dean F. Jerding, John T. Stasko, Thomas Ball

May 1997 **Proceedings of the 19th international conference on Software engineering**

Full text available: pdf(1.72 MB) Additional Information: full citation, references, citings, index terms

Keywords: object-oriented software engineering, program understanding, reverse engineering, software visualization

25 Anatomy of a hardware compiler

K. Keutzer, W. Wolf

June 1988 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1988 conference on Programming Language design and Implementation**, Volume 23 Issue 7

Full text available: pdf(982.45 KB) Additional Information: full citation, abstract, references, citings, index terms

Programming-language compilers generate code targeted to machines with fixed architectures, either parallel or serial. Compiler techniques can also be used to generate the hardware on which these programming languages are executed. In this paper we demonstrate that many compilation techniques developed for programming languages are applicable to compilation of register-transfer hardware designs. Our approach uses a typical syntax-directed translation → global optimization → local ...

26 A fast and accurate framework to analyze and optimize cache memory behavior

Xavier Vera, Nerina Bermudo, Josep Llosa, Antonio González

March 2004 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 26 Issue 2

Full text available: pdf(270.06 KB) Additional Information: full citation, abstract, references, index terms

The gap between processor and main memory performance increases every year. In order to overcome this problem, cache memories are widely used. However, they are only effective when programs exhibit sufficient data locality. Compile-time program transformations can significantly improve the performance of the cache. To apply most of these transformations, the compiler requires a precise knowledge of the locality of the different sections of the code, both before and after being transformed.Cache ...

Keywords: Cache memories, optimization, sampling

27 Symbolic pointer analysis

Jianwen Zhu

November 2002 **Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design**

Full text available: pdf(111.91 KB) Additional Information: full citation, abstract, references, index terms

One of the bottlenecks in the recent movement of hardware synthesis from behavioral C programs is the difficulty in reasoning about runtime pointer values at compile time. The pointer analysis problem has been investigated in the compiler community for two decades, which has yielded efficient, polynomial time algorithms for context-insensitive analysis. However, at the accuracy level for which hardware synthesis is desired, namely context and flow sensitive analysis, the time and space complexit ...

CiteSeer Find: analyzer vertex and control flow Documents Citations

Searching for **analyzer vertex and control flow**.
Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Google (CiteSeer) Google (Web) CSB DBLP
No documents match Boolean query. Trying non-Boolean relevance query.
500 documents found. **Order: relevance to query.**

On the Limit of Control Flow Analysis for Regression Test Selection - Ball (1998) (Correct) (13 citations)
at compile time. In P's **control flow** graph G, each **vertex** represents a basic block of instructions and each
On the Limit of **Control Flow** Analysis for Regression Test Selection
On the Limit of **Control Flow** Analysis for Regression Test Selection Thomas
www.bell-labs.com/~tball/papers/issta98.ps.gz

System-Dependence-Graph-Based Slicing of Programs with.. - Sinha, Harrold (1998) (Correct) (15 citations)
or expression depends. Each PDG contains an entry **vertex** that represents entry into the procedure. To
Slicing of Programs with Arbitrary Interprocedural **Control Flow** Saurabh Sinha Computer and Info. Science
www.cis.ohio-state.edu/~harrold/research/./webpapers/icse99.ps

Loop Distribution with Arbitrary Control Flow - Kennedy, McKinley (1990) (Correct) (15 citations)
Loop Distribution with Arbitrary **Control Flow** Ken Kennedy Kathryn McKinley CRPC-TR90064
Loop Distribution with Arbitrary **Control Flow** Ken Kennedy Kathryn McKinley CRPC-TR90064 August
`90. Loop Distribution with Arbitrary **Control Flow** Ken Kennedy Kathryn S. McKinley Rice University
softlib.rice.edu/pub/CRPC-TRs/reports/CRPC-TR90064.ps.gz

The Execution Order Control Method among Coarse Grain.. - Koichi Asakura (Correct)
The Execution Order **Control** Method among Coarse Grain Processes on
algorithm are introduced, first of all. **Control flow** graph[15] Our **control flow** graph G =V E) is a
first of all. **Control flow** graph[15] Our **control flow** graph G =V E) is a directed acyclic graph and
fcapwide.fujitsu.co.jp/pcw/pcw95j/p2c.ps.gz

Static Analysis of Processes for No Read-Up and No.. - Bodei, Degano, Nielson.. (1999) (Correct) (19 citations)
at a lower level. The static check is based on a **Control Flow** Analysis for the calculus that
lower level. The static check is based on a **Control Flow** Analysis for the calculus that establishes a
security properties. We use here **Control Flow** Analysis that is a static technique for predicting
www.di.unipi.it/~chiara/publ-40/BDNN99.ps

Intelligent Computing About Complex Dynamical Systems - Zhao (1994) (Correct)
with the system. MAPS, standing for Modeler and **Analyzer** for Phase Spaces, is an autonomous phase-space
mechanisms for intelligently simulating nonlinear **control** systems. These mechanisms enhance numerical
www.cis.ohio-state.edu/insight/papers/mcs.ps

Generating Analyzers with PAG - Alt, Martin, Wilhelm (1995) (Correct)
Generating **Analyzers** with PAG Martin Alt Florian Martin Reinhard
domain functionality nor with the traversal of the **control flow** graph or syntax tree nor with the
www.cs.uni-sb.de/RW/users/martin/techpag.ps

The Machine SUIF Bit-Vector Data-Flow-Analysis Library - Holloway (1998) (Correct)
for iterative, bitvector -based data-**flow analyzers**. It uses Machine SUIF's **control-flow** graph
-based data-**flow analyzers**. It uses Machine SUIF's **control-flow** graph (CFG) library [4] to parse the
The Machine SUIF Bit-Vector Data-**Flow**-Analysis Library Glenn H. Holloway Allyn Dimock
www.eecs.harvard.edu/~hube/machsuif2-doc/bvd.ps

Efficient Path Profiling - Ball, Larus (1996) (Correct) (80 citations)
directed acyclic graphs (DAG) with a unique source **vertex** ENTRY and sink **vertex** EXIT .Section 4 shows
frequencies, shown by the edge frequencies in the **control-flow** graph. In path profile P rof1, path ABCDEF
shown by the edge frequencies in the **control-flow** graph. In path profile P rof1, path ABCDEF is
www.stanford.edu/class/cs343/ps/pathprof.ps

Automatic Data Decomposition for Message-Passing Machines - Damian-Iordache, Pemmaraju (1997) (Correct) (1 citation)
distributions, array replications and **control flow** are taken into account. This paper
distributions, array replications and **control flow** are taken into account. This paper formulates the decomposition for a program with general **control flow** induced by if-statements and loops. The automatic
ftp.cs.uiowa.edu/pub/sriram/graphs/lcpcTr.ps.gz

Program Dependence Graphs for the Rest of Us - Ballance, Maccabe (1992) (Correct) (8 citations)
purposes, every PDG is rooted at a distinct **vertex** named Entry. We assume every **vertex** in a PDG is
Inc. Abstract This report presents new **control** dependence analysis techniques that succeed in
ftp.cs.unm.edu/pub/cs_tech_reports/1992/CS92-10.ps.Z

A Partial Approach to the Problem of Deadlocks in.. - Tricas.. (1998) (Correct)
October 28, Abstract In the modeling and **control** of manufacturing systems one of the aims is to
set W :P \Theta T)T \Theta P)IN is the **flow** relation: if W (x y) 0, then we say that there
represented as N =hP T Ci where C is the net **flow** matrix: a P \Theta T integer matrix so that C =C
www.cps.unizar.es/~ftricas/GISIRR9705.ps.gz

A TMS320C40 based Speech Recognition System for Embedded.. - Obermaier, Rinner (1998) (Correct)
sentences [8]On the other hand, command and **control** applications [2] recognize only a few specific
www.iti.tu-graz.ac.at/en/people/rinner/../../publications/papers/obermaier98.ps.gz

Complete Removal of Redundant Expressions - Bodik, Gupta, Soffa (1998) (Correct) (10 citations)
framework, we derive a demand-driven frequency **analyzer** whose cost can be **control**led by permitting a
are strictly partial. To achieve a complete PRE, **control flow** restructuring must be applied. However, the
partial. To achieve a complete PRE, **control flow** restructuring must be applied. However, the
www.cs.pitt.edu/~soffa/research/Comp/pldi98.ps

Data Flow Analysis for Reverse Engineering - Moonen (1996) (Correct) (17 citations)
: 2 1.2.1 **Control flow** normalization :
www.wins.uva.nl/pub/programming-research/reports/1996/P9613.ps.Z

Optimally Profiling and Tracing Programs - Ball, Larus (1992) (Correct) (106 citations)
graph representation of a program, where a **vertex** represents a basic block of instructions and an
in Well-Delimited Programs 23 6.3 Profiling Using **Control** Dependence 23 6.4 Optimal Placement of Traversal
www.deas.harvard.edu/cs/academics/courses/cs248r/readings/qpt-opt-prof-trace.ps.gz

Predicting Data Cache Misses in Non-Numeric Applications.. - Mowry, Luk (1997) (Correct) (19 citations)
lifetimes which can lead to spilling, and software-**control**led prefetching requires additional instructions
correlated with information such as recent **control-flow** paths, the cache outcomes of previous references,
to distinguish these contexts. The first is **control-flow** information-i.e. the sequence of N basic block
www.cs.cmu.edu/~luk/luk_papers/micro97.ps.gz

The Architecture, Operation and Design of the Queue Management.. - Kozyrakisy (1996) (Correct)
multiple classes of service, multicasting and **flow control**, enforce further extensions to the above scheme
as multiple classes of service, multicasting and **flow control**, enforce further extensions to the above
for multicasting and multi-lane credit-based **flow control**. Techniques such as pipelined and
www.ii.uib.no/~markatos/avg/papers/1996.TR172.QueueManagement.ps.gz

Credit-Flow-Controlled ATM versus Wormhole Routing - Katevenis, Serpanos, Spyridakis (1996) (Correct)
FORTH-ICS /TR-171 July 1996 Credit-**Flow-Control**led ATM versus Wormhole Routing M. Katevenis, D.
FORTH-ICS /TR-171 July 1996 Credit-**Flow-Control**led ATM versus Wormhole Routing M.
networks. When ATM is combined with credits-the **flow control** mechanism that is particularly suitable
www.ii.uib.no/~markatos/arch-vlsi/papers/1996.TR171.ATM_vs_Wormhole.ps.gz

Towards a Crystal Ball for Data Retrieval - Hellerstein (Correct)
can predict the utility of their query results, **control** the behavior of the queries on the fly, and
s2k-ftp.cs.berkeley.edu/postgres/papers/ngits97-control.ps.Z

First 20 documents Next 20